

経過報告

田中 隆己

2011年1月14日

1 注釈

TDC calibrator が来たので軽く 1,2ch だけ測定した。これより、今まで 5% ほど時間を短く見積もっていたことがわかった。よって、以下の時間についての結果に消費税を加えたものがおよそ正しい結果となる。測定に区切りがいたら再度 TDC calibration を行う予定である。

1.1 まとめ

やったことは、

- smallNEUT 評価実験
 - 回路全チェック
 - * visalscaler に問題発覚
 - * 1ch が放電してる模様
 - 回路図のメモをとった (電子化してない)
 - GAIN 調整, エネルギーキャリブレーション (SOLDIRAs)
 - thru 補正
 - 時間分解能、時間分解能のエネルギー依存
 - 位置キャリブレーション (7ch,8ch の NEUT は不正確)
 - アテネーションのパラメータを求めた。
- rdfconverter 作成 (txt to rdf, rdf to txt))
- fortran で c を扱う (enc_bbdaq.f の解析部分を c でかけるようになった)
- DEMONS を anapaw で解析 (raw はちゃんと見えたので、まずは運動量のスペクトルを出す)。

やらなければいけないことは、

- smallNEUT 評価実験
 - 時間分解能のエネルギー依存試しに 3 日分の run で解析したが、peek のエネルギーから外れるほど (高エネルギー側も) 時間分解能が落ちてしまった。また、低エネルギー側について \sqrt{e} でフィットしたが、切片がほぼ 0 となってしまう、低エネルギー側も期待される時間分解能より悪くなっていることが分かった。trigger を斜めにしてみたが、これではイベントがたまらなさ過ぎたので却下。しかたがないので、2*4 を 4*2 にして測定してみたいと思う。もしくは、trigger をずらすの

ではなく、neut を斜めにするとか。

- DEMONS 卒論では DEMONS の結果がかなりを占めるかもしれないので、DEMONS の結果のまとめをしなければならない。しかし、DEMONS によるシュミレーションの解析は、直接コードに埋め込んでいて、既に収集がつかなくなっている。このため、解析を外部化したかった。すでに外部化する手はずは整ったので、時間の限り、結果の再導出をしたい。また、これはのちの geant4 の結果のチェックにもつながると考えている。
- Geant4 によるシュミレーションこれが卒論の主題にしたいのだが、物理を登録するところが手つかず。おそらくハドロン相互作用を入れてもうまく行かないので、自分で、プロトンだけの物理を入れたり、DEMONS を移植したりしなければならない。これがどの程度時間がかかるか未知数。また、現在の構成ではお金がかかりすぎるので、ガス検出器を用いた場合のシュミレーションもしたい。これは完全に構想のみ。
- 卒論を書かなければいけない。物理背景とか、物理その物がまだ分かっていないことがたくさんあるので、かなり勉強しなければいけないと思われる。
- 具体的にどの物理を目指すかを考えるために、勉強しなければならない。
- アクセプトランスは計算しなくてはいけない。そのために、フェルミ運動量とビームの広がりについて理解しないとイケない。

2 評価実験周り

2.1 回路チェック

回路の全チェック、といっても機能を完全にチェックしたわけではなく、大まかに正しく動作しているか、配線が間違っていないか等のレベルでチェックした。

2.1.1 visualscaler

visualscaler の input から 1GHz、全幅で数十 mV の sin 波が出ていた (最大 70mV、最小 25mV、平均 50mV 程度)。これは時間分解能に影響するので (100ps 程度)、気をつけなければいけない。独立な経路に接続して対処。

2.1.2 trigger 用の PMT:1ch

1ch の PMT が時々放電している様であった。とりあえずは coincidence を取れば問題にならないので放置した。

2.1.3 反射波

ディスクリがたまに反射波を拾ってしまい、二連続で discr1 からシグナルがでてしまうことがあることがわかった。discr1 の width を広げて対処。

2.2 回路図

時間があればここに回路図がはられているはず。卒論で必要なのでいずれにせよ書く。

2.3 バックグラウンド

バックグラウンドに gamma 線があることが分かった。エネルギーの最大値は 2.5MeV 程度。これにゲートをかけて上下の時間差を見てみると、山が二つ見え、上から下と、下から上へ行く二つの gamma と解釈できる。この gamma 線はタイミングが違うので (と言うか、経路誤差が大きいので) 時間分解能の測定には使えない。よって 2 本の NEUT の coincidence を取るときは gamma 線付近は捨てて解析した。

この gamma 線が何の由来かは調べていない。(何の本を見ればバックグラウンドの gamma が分かるのだろうか?)

2.4 GAIN 調整, エネルギーキャリブレーション (SOLDIRAs)

gain の fit は、どうしても $a \cdot HV^k$ とならなかったため、対数軸上で、多項式近似をした。これについて物理的な解釈はできていないが、gain 調整のためなのでよしとした。

ch to MeV は SOLDIRAs によって求めた peek のエネルギーを使った。この結果、NEUT の peek は 3.50MeV、trigger の peek は 1.68MeV と求めた。

HV の設定は NEUT は 250keV から 50MeV まで取れるようにした。また、Trigger は peek が 500ch になるようにした。

ハードの調整では gain が正確に合わなかったため、ソフトで微調整した。

2.4.1 パラメータ

GAIN 調整が終わったので、HV と thr を示す。

	HV[V]	thr[mV]
1ch	-1504	-50.7
2ch	-1757	-50.4
3ch	-1425	-50.7
4ch	-1360	-50.6
5ch	-1275	-50.6
6ch	-1362	-50.6
7ch	-1334	-50.4
8ch	-1189	-50.5

表 2.1 run1049 以降の HV と thr

2.5 スルー補正

2.5.1 スルー補正について

前回、スルー補正するのに、横軸を両チャンネルの相乗平均でプロットするとよい理由を示したが、それを電子化しておく。

スルー $= C_1 / \sqrt{A} + C_2$, $A = A \exp(-z/\lambda)$ を仮定する。スルーは二つの PMT の時間積和 $(T_a + T_b)$ ごとに考

えるので、

$$\frac{1}{(A_a A_b)^{1/4}} \quad \text{でプロットするとき} \quad (2.1)$$

$$T'_a + T'_b = T_a + C_{1a}/\sqrt{A_a} + C_{2a} + T_b + C_{1b}/\sqrt{A_b} + C_{2b} \quad (2.2)$$

$$= T_a + T_b + \frac{1}{(A_a A_b)^{1/4}} \left(C_{1a}(A_b/A_a)^{1/4} + C_{1b}(A_a/A_b)^{1/4} \right) + C' \quad (2.3)$$

$$(2.4)$$

$$C_{1a}(A_b/A_a)^{1/4} + C_{1b}(A_a/A_b)^{1/4} = C_{1a} \exp\left(\frac{z_a - z_b}{4\lambda}\right) + C_{1b} \exp\left(\frac{z_b - z_a}{4\lambda}\right) \quad (2.5)$$

$$\therefore z \ll 4\lambda \text{ のとき} \rightarrow C_{1a} + C_{1b} \quad (2.6)$$

$$\frac{1}{\sqrt{A_a}} \quad \text{でプロットするとき} \quad (2.7)$$

$$T'_a + T'_b = T_a + C_{1a}/\sqrt{A_a} + C_{2a} + T_b + C_{1b}/\sqrt{A_b} + C_{2b} \quad (2.8)$$

$$= T_a + T_b + \frac{1}{\sqrt{A_a}} \left(C_{1a} + C_{1b}(A_a/A_b)^{1/2} \right) + C' \quad (2.9)$$

$$(2.10)$$

$$C_{1a} + C_{1b}(A_a/A_b)^{1/2} = C_{1a} + C_{1b} \exp\left(\frac{z_b - z_a}{2\lambda}\right) \quad (2.11)$$

$$\therefore z \ll 2\lambda \text{ のとき} \rightarrow C_{1a} + C_{1b} \quad (2.12)$$

となる。実際は、どちらの方法でやっても、位置の不確か性が小さいとき、あるいは中心で測定したときはスルーは左右の和となる。さらに、スルーの個体差が小さいならば、 $(T_a + T_b)/2$ のフィットの結果がそのまま一つ分のスルーになることがわかる。

2.5.2 スルー補正の結果

fit は paw の hist/fit で線形 fit した場合と、目で線形 fit した場合を用いた。fit はかなり誤差を含んでいたため、スルーは同一のモジュール (1ch2ch3ch4ch or 5ch6ch7ch8ch) について平均値を取った。(thr を 50.4 50.8mV に揃えてあり、Trigger の PMT は型番は違うが H7195 は H1161 の後継機で同じもの、NEUT は個体差を除いて同じもの。HV はさほど影響しないと考えた。)

次に、求めたスルーを使い、時間分解能を出し、それぞれの検出器について一番よい時間分解能が得られたものをスルーとした。また、二つの独立に求めた時間分解能 (Trigger を使う or 二本の NEUT の sqrt) が consistent であることを確かめることにより、スルーがおかしくないことを確認した。

スルーの結果を図 2.1 に示す。

上で求めたように、今回はシンチレータの中心で測定しているので、位置の誤差は現れず、一回の fit で十分正しいスルーが得られたと考えられる。ただし、エネルギーが低い方、高い方どちらも時間が遅くなる傾向にあり、また時間分解能も悪かったので、peek 以外は経路の誤差などのスルー以外の要素が効いていると考えられる。これについては特に補正せず、経路に対して厳しく gate(12345678) をかけることによって、時間が遅くなっているものを間接的に除いた。

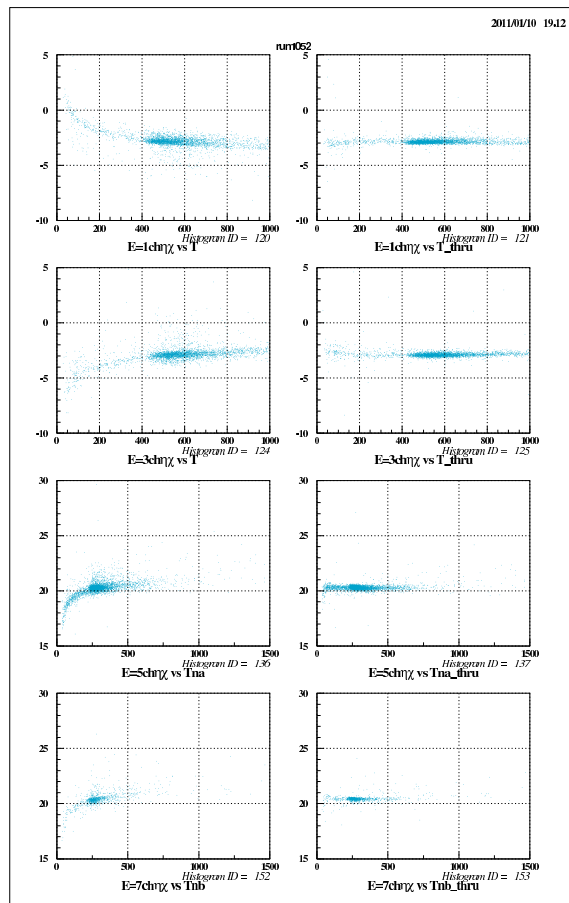


図 2.1 thru 補正の結果

2.6 時間分解能

GAIN も決まり、スルー補正もできたので、改めて時間分解能を出した。コロキウム第 2 より前の結果はザックリだったのに対して、今回の結果は GAIN とスルーが比較的正確なので信頼できる。また、12345678gate を取ることによって経路の誤差を極力減らし、純粋な時間分解能を得ることを目指した。(ただし、回路自体の時間分解能の評価はまだ)

結果を表 2.2 に示す。

T_n と T から時間分解能を求めると、(PMT 二つで)70ps 程度となったが、これは Trigger の時間分解能が 95ps 程度なので信頼性にかける。幸いにも、今回は同一の NEUT が二本あるので、これらからも時間分解能を求めることができ ($T_{ab}/\sqrt{2} = ?$ T56, T78 or $\sqrt{\sigma_{T56}^2 + \sigma_{T78}^2} = ?$ T_{ab})、consistent な結果が得られたので、70ps という数字は信頼できるといえる。

印象として、12345678gate をかけると 3 割 (30ps) ぐらい時間分解能があがるのでここまでくると、経路誤差が重要になってくるのではと考えられる。今回は trigger のサンドイッチのさらに上に NEUT が置いてあるので、より経路を細く絞ることができたのがよい結果につながったと思う。

以上から、時間分解能は NEUT 一本あたり $\sigma < 70ps$ (@3.50MeVee) であると結論づける。時間分解能のエ

[ps]	before	after
T	332.70	131.30
Tna	315.10	94.81
Tnb	270.90	95.56
T56	267.61	68.40
T78	213.81	69.44
Tab	279.50	97.83
Tab/ $\sqrt{2}$	197.64	69.18

表 2.2 run1051,52,53, center(50cm), gate:12345678

エネルギー依存は次で議論する。

2.7 時間分解能のエネルギー依存

エネルギーごとの時間分解能を出すために、得られたデータを分割してみたが、期待どおりにいかず、宇宙線のピークを底にして、それ以外のエネルギーでは時間分解能が悪くなってしまった。(具体的には $\sqrt{5*6}$ について 0.5MeV ごとに分割して、そのシグマを求めた。Tna の場合はいつもの $\sqrt{Tna^{**2}-T^{**2}/4}$ で計算し、Tab の場合は $Tab/\sqrt{2}$ で計算した。この計算はそれぞれのもつ相関に依存してしまうので、どちらも近似でしかない。)

この理由として、一つは thru 補正をしても、高低エネルギーが補正しきれていないことがあげられる。補正できない分は経路誤差かとも思ったが、説明がうまくつかない。

また、低エネルギー側は信頼できると思って低エネルギーのみ $a/\sqrt{E}+b$ でフィットしたが、切片が 0 近傍になってしまった。エネルギーが上がると時間分解能が 0 近傍に収束するとは考えにくいので、低エネルギーも統計誤差やスルー以外の要因で分解能が悪くなっていると考えられる。

とにかく、このままでは議論できないので、シンチレータを縦 (4cm を上) にして今測定している。(トリガーを斜めにするのはレートが下がりすぎたので諦めた) これにより、高い精度で二倍のエネルギーを狙えると考えている。また、エネルギーの低い方に関しては、細いシンチレータから間接的に求めようかと考えている。

その後、角度がついた場合は発行する位置が PMT に対して変化するのでそれによる時間のブレが影響しているのではという予想が立った。これを検証するために、NEUT を斜めにして測定することを検討している。

また、宇宙線の落とすエネルギーを変化させずとも、アテネーションを利用して擬似的にハイエナジーないしローエナジーのシグナルの時間分解能を評価できるはずなので、これも解析してみる。

現在は 4cm を縦にした 2 本の NEUT について測定していて、gate をかなり絞った (宇宙線の角度は ± 15 度、シンチレータの位置は中心の 15cm の幅) 結果、NEUT 一つ分の時間分解能は 59.8ps と求まった。また、run1049,61(2cm を縦にした) について同様の解析をしたら 63.1ps と求まった。(これが前の時間分解能の項目と一割ずれているのは現在検討中。本来ならこの gate は前の解析より緩い gate のはずなので、悪くなると直感的には考えられることと反する。) この二点から $1/\sqrt{E}$ の係数が求まるので、誤差を考えずに当てはめたとこ、(時間分解能 [ps]) = $20.8/\sqrt{\text{エネルギー [MeV]}} + 52.0$ と求まった。先の結果とは違って、ごく自然なパラメータとなった。これだけではまったく信頼性がないので、これを結論とはできないが、高エネ

ルギーおよび低エネルギーで時間分解能が悪くなっていたことは、宇宙線や光子の経路、解析方法の問題であることが示唆された。

2.8 位置キャリブレーション、アテネーション

2.8.1 位置キャリブレーション

センターをだして、光の伝搬速度から左右の位置を出したが合わなかった（これは消費税文かもしれないがチェックしていない）。そのため位置キャリブレーションを行なった。三点でキャリブレーションした。結果は、

$$5, 6ch : (\text{位置 [cm]}) = 15.85(\pm 0.03) * (\text{時間差 [ns]}) - 6.02(\pm 0.05) \quad (2.13)$$

$$7, 8ch : (\text{位置 [cm]}) = 15.71(\pm 0.01) * (\text{時間差 [ns]}) + 9.70(\pm 0.03) \quad (2.14)$$

$$(2.15)$$

となった。それぞれの偏差の二乗平均は 0.5mm と 0.3mm となり、十分正確なキャリブレーションができた。

2.8.2 attenuation の予測

後藤さんにマイラーの反射率を聞いたが返事が帰ってこない。マイラーの反射率が分からないとパラメータが 4 つ (キャスト面の反射率 (Rb)、研磨面の反射率 (Ra)、マイラーの反射率 (Rmylar)、シンチレータの純粋な減衰長 (λ)) で、且つそれぞれが似たような働きをするので、正確に求めるにはマイラーではなく黒いシートで覆うなどして、パラメータを減らして実験をする必要がある。

それを踏まえて、とにかく解析してみた。geometry は、二本の NEUT を平行に置いた。解析は、位置対エネルギーの二次元ヒストグラムを 5cm ごとに区切り、それぞれについて宇宙線のピークを求めた。(このとき、端の方はイベントが少ないので正確には補正が必要だが、今回は無視した。) この実験結果に対して、前に導出したアテネーションの式 (2.16) により fit した。

$$Attenuation = \frac{1}{2\pi} \iint R_a^{[|z \tan \theta \cos \phi / b|]} R_b^{[|z \tan \theta \sin \phi / a|]} \exp\left(-\frac{z}{\lambda \cos \theta}\right) \sin \theta d\theta d\phi \quad (2.16)$$

このとき、式が積分関数であることと、パラメータが 4 つあり自由度が大きいことから、いくつかのパラメータごとに誤差を求め、最小となるところを大雑把に求めてから、その周囲で極小になるところを総当たりで求めた。

これより、最適なパラメータは Ra=0.994, Rb=0.993, Rmylar=1.000, $\lambda=950$ [cm] と求まった。この結果を図 2.2, 2.3 に示す。

また、誤差の二次元プロットを図 2.4, 2.5, 2.6 に示す。

誤差の様子から分かるように、はっきりとした極値はなく、それぞれのパラメータが一定の関係を満たすところは十分解になりうるので、今回求まったパラメータが真の値とはいきれない。(mean の誤差は小さいが、sigma が大きいといえる。)

Ra, Rb は 0.98 から 1.000 の範囲で、Rmylar は 0.99 から 1.000 の範囲で、減衰長は 500cm から 3000cm 程度の範囲でカイが 2% 以下程度の適当なパラメータをとることができる。

また、どのようなパラメータをとってもカイ二乗が 0.01 を割らないので、1% の誤差を内在していると言える。

期待としては、Ra \ll Rb となる (明らかにキャスト面の方が可視光をよく反射していた) 予定だったので、

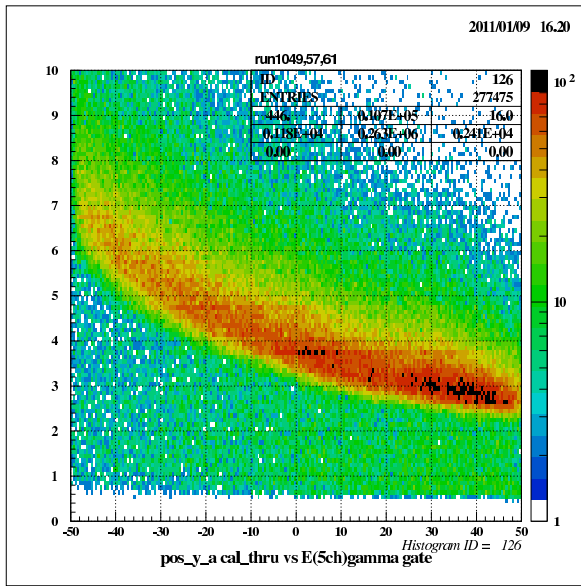


図 2.2 attenuation, 実験結果 (run1049,57,61, 5ch(左下))

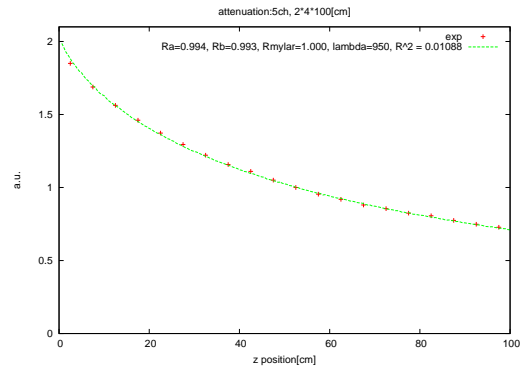


図 2.3 attenuation, fit 結果

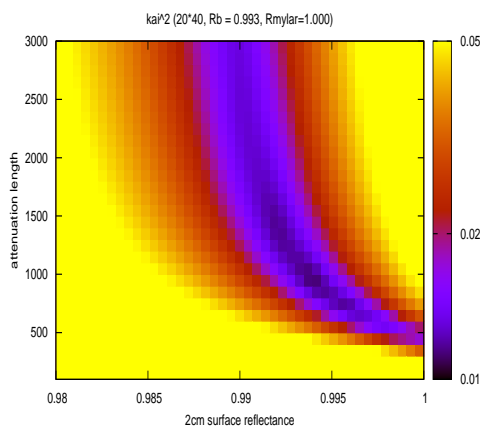


図 2.4 Ra vs λ (Rb=0.993, Rm=1.000 に固定)

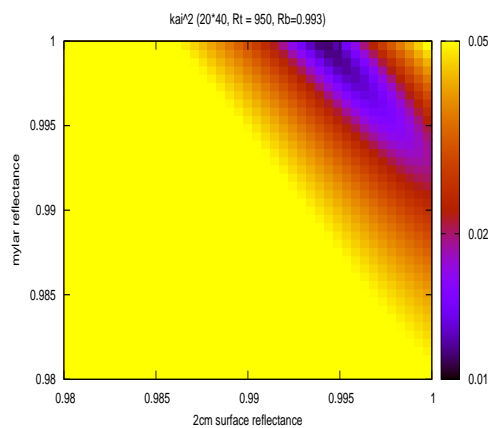


図 2.5 Ra vs Rmylar (Rb=0.993, λ =950 に固定)

直感と反する ($R_a \approx R_b$) 結果になった。しかし、アテネーション自体は 1% 程度の誤差でフィットできたので、式の正しさが示せた。式の補正は規格化だけなので、十分有用な式が導出できたのではと考えられる。この後は細いシンチレータを実験していくので、さらに妥当性を確かめていく。

3 VME モジュールなど

VME モジュールなどの購入について。

- ・VME クレート

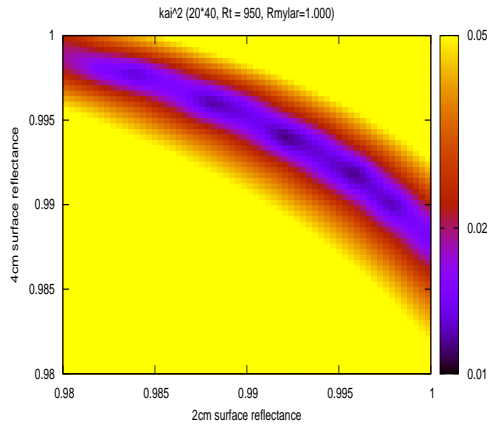


図 2.6 Ra vs Rb($\lambda=950, R_m=1.000$ に固定)

CAEN と WIENER では WIENER のほうが安かった。電源容量は WIENER の方が大きくて安い。また、WIENER には種類が豊富。

- ・HV クレート

初期投資は WIENER のほうが安い、ch あたりでは CAEN の勝ち。CAEN は HV クレートにコンピュータが入っているのでローカルに調整しやすいが、WIENER は外部からじゃないと操作できないので不便かも。また、マニュアルを読むと外部からの操作にプログラムを書かなければいけない様に見えた。

- ・VME コントローラ

- ・618-3

EOL がかかってしまい、3/31 までの受注となった。枯れたコントローラだから一つは持っておくといいかも。

- ・VM-USB は期待の新生。

usb なのでノートパソコンでも扱えるかも。さらに、速度が出しやすい仕様らしい。1 2 月に馬場さんが入手するそうなので、その経過待ち。計画としては、VM-USB の結果を見極めてから、コントローラを買う。VME モジュールとの時差は 618-3 を NEBULA から借りることにより補う。

QDC、TDC、ディスクリなどは CAEN にしかない。(正確には WIENER の TDC は納期 6 ヶ月、QDC はなく FLASH ADC ならある)

計 500 万円弱。

4 水素 5 の測定

物理的な背景などはまだ勉強不足。卒論までには調べる。

- ・レート計算

$10^{*}6$ 個/s の ${}^6\text{He}$ がくると仮定する。プロトンノックアウトの断面積が 1mb として、 $2\text{g}/\text{cm}^2$ の $C(A=12)$ ターゲットとする。 $\text{barn}=10\text{e-}24$ なので、 $\text{rate} = (1\text{mb} * 2\text{g}/\text{cm}^2 * 6.0 * 10\text{e+}23) / A = 10\text{e-}4$ ${}^6\text{He}$ のレートを

かけると、 $10e+2 s^{-1}$ できる。1 時間で 100 個イベント欲しければ、3600 分の 1 まで efficiency が落ちててもよい。2n なので、一つあたり 60 分の 1 となる。プロトンの解析可能イベントが多くても 4 分の 1 程度。

1m*1m のときにアクセプタンスはざっくり 2 分の 1 とする。このとき、中性子検出器の厚みは 15cm となる。これだと 320ch(160 本) 必要になり、1ch あたり 23 万円ぐらいなので、モジュールと回路で 6000 万強。

10 時間で 100 イベントとすると、6cm で足りることになり、これを満たすには 4 段必要になる。これだとプロトンのトラッキングがどこまでできるかわからないが、160ch(80 本) なので 3200 万強。

架台を考えると、いずれにせよお金が足りない。アクセプタンスを絞って 40cm*40cm*8cm だとしても、80ch(40 本) で 1600 万強。これでも回路を買わないという選択肢を選ばない限り実現できない。

ここで、40cm*40cm がどれだけいけるかについて考察した。

アクセプタンスのざっくり計算。相対エネルギーによって一番広がる場合を古典論で計算する。二粒子の場合、進行方向に対して直角に崩壊した場合に角度が比較的大きくなる。この時、内角は $\theta = \theta_1 + \theta_2$, $\tan \theta_1 = v_1/v$, $\tan \theta_2 = v_2/v$ で表される。相対エネルギーは質量の逆比で分配されるので、 $\tan \theta_1 = \sqrt{(E_{rel} * A_2 / (A_1 + A_2)) / E}$, $\tan \theta_2 = \sqrt{(E_{rel} * A_1 / (A_1 + A_2)) / E}$ 2 つ中性子のうち、一つの中性子にエネルギーが集中したとすると最大の角度になり、 $\theta_{max} \simeq 5/100$ となった。つまり、10m で 50cm のアクセプタンスがあれば相対エネルギーについては 3MeV までとれる。2MeV に対してだと 10 で 40cm となる。これに、フェルミ運動量によるブレと入射粒子のブレが入ってくるので、10m で 40cm ではアクセプタンスが 1/2 も取れないかもしれない。

段数も少ないのでおもしろくない。

5 rdfconverter

rdfconverter(txt to rdf(rdf to txt もできるけど不完全)) を作った。動作検証としてわかりやすいデータ列で検証した上で、SOLDIRAs で吐いたデータに対して、anapaw の gate を使って解析をし、物理的に正しい結果が得られた。これにより、モンテカルロシミュレーションにおいてコード内で解析せずに、単純に可読な txt で 1 イベントごとにデータを吐いて、anapaw で解析することができるようになった。

シミュレーションじゃなくても使い道はあるかもしれない。

6 fortran で c を扱う

fortran で c を扱うポイントが分かったので示す。これを受けて、とりあえず enc.bbdaq の解析部分を c で書くことに成功した。(というか、重要なのは val,naok が c の関数にちゃんと渡せて、c のなかで val と naok をつめていくことができた。) ただ、cpp はどうもできなかった。暇があったら、enc.bbdaq.f をまるごと c で扱えるようにしたいなと思っている。

以下、fortan と c のやりとりの仕方。

関数の名前は、c では func_(int* a) となり、f では func(a) となる。c で関数の名前にアンダーバー二本を付けるという規約。また、fortran の関数の変数は必ず参照渡しになるので c ではポインタとして受け取る、もしくはアドレスを渡す。

なので、その値を参照したり、その変数に代入するには

*a = 1.0, とか、int num = *a

とか。あとは、配列についてで、fortran で a(5) と宣言したら、a(1),a(2),...,a(5) だが、c では a[0],a[1],...,a[4]

とゼロから始まる。また、二次元配列は fortran で $a(5,5)$ と宣言したら、 $a(1,1), a(2,1), a(3,1), \dots, a(5,5)$ という順番なのが、c では、 $a[0][0], a[0][1], a[0][2], \dots, a[5][5]$ というふうに順番が逆になる。これが重要なのはメモリの確保の順番で、この順番でメモリが確保されていること。配列を関数間でやりとりするのに、関数に直接配列を渡すことはできないので、配列の先頭の値のアドレスを渡す。この先頭アドレスを元に連なる多次元配列にアクセスするためには、アドレスの足し算をするのだが、この時メモリの順番が重要になってくる。

具体的には、fortran で

```
Integer val(2,3)
```

と宣言したら、c の関数 func を

```
func(val(1,1), 2, 3) //func(先頭のアドレス, 一列目の配列の要素数, 二列目の配列の要素数)
```

で関数を呼び、c では、

```
func(int* val, int* nx, int* ny)
```

```
{  
    // val[2][1] = 5 // f で言う val(2,3) にアクセスするには直感ではこうなるが、これは不可。  
    // (ちなみに c では一個数字が減るので 2 が 1 で 3 が 2 で逆順なので (2,3) [2][1])  
    *(val + 2*(nx) + 1) = 5 // こうやってアドレス演算する  
}
```

という感じで受け取る。これでうまくいくので、(もしかしたら nx じゃなくて ny かも) これを適当な形で define してしまえば意識しなくとも配列にアクセスできる。

common 文は extern で取れるらしいが試していない。

コンパイルは個別にコンパイルして、g77 とかでリンクすればよい。

```
gcc -c func.c
```

```
g77 -c func.f
```

```
g77 -o func func.o func.f.o
```

とかでいけた気がする。

これで、user が書くところは DEBUG がしやすい c に簡単に移行できる。

7 DEMONS

rdconverter と c 化した enc ができ、demons のシュミレーション結果を簡単に解析できる環境が整ったので、demons の結果を再導出しようと考えている。これは、geant での結果と比較するためにも必要と考えている。また、geant の出力を root ではなく anapaw で、同じ enc、anafire で解析できるのではと期待している。

現在、raw はちゃんと見えたので、まずは運動量のスペクトルを出すことを目指す。とりあえずいじり始めてわかったことが、時間が謎にシフトしていて、これが前に mean が合わなかった原因と思われる。ソースを追っかけたが読めたものではなかったので、強制的に補正した。

8 細かい memo

- ついに 554 の水道の下水溝から虫が発生してしまった。桶をかぶせてスプレーで抹殺したが、根本的解決になっていない。